PATENT ABSTRACTS OF JAPAN

(11)Publication number : 05-225149

(43)Date of publication of application : 03.09.1993

(51)Int.Cl. G06F 15/16
G06F 9/46
G06F 12/00

(21)Application number : 04-025619

(22)Date of filing : 13.02.1992

(71)Applicant : TOSHIBA CORP

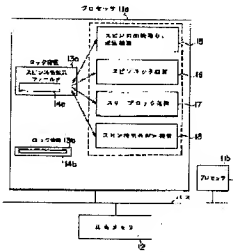
(72)Inventor : KITSU TOSHIKI

(54) LOCK SYSTEM

(57)Abstract:

PURPOSE: To improve the efficiency of process execution by reducing the wasting of a processor at the time of lock acquisition.

CONSTITUTION: When the lock of a lock variable 13a can not be acquired, the process is set in a spin lock state by a spin lock mechanism 16 and when the lock can not be acquired even after a specific spin lock time spin, a sleep lock mechanism 17 sets the process in a sleep state. Consequently, the process is put in the sleep state unless the lock is obtained for a time exceeding the specific spin lock time set in a spin time indication field, and the processor is passed to another process; when the lock is released from another processor relatively early, the lock is acquired since the process is in the spin lock state, so that the process can be executed.



LEGAL STATUS

[Date of request for examination]

[Date of sending the examiner's decision of rejection]

[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]

[Date of final disposal for application]

[Patent number]

[Date of registration]

[Number of appeal against examiner's decision of rejection]

[Date of requesting appeal against examiner's decision of rejection]

[Date of extinction of right]

(19)日本国特許庁（ＪＰ）　(12)**公開特許公報**（Ａ）　(11)特許出願公開番号

特開平5－225149

(43)公開日　平成５年(1993)９月３日

(51)Int.Cl.[5]	識別記号	庁内整理番号	Ｆ Ｉ	技術表示箇所
Ｇ０６Ｆ　15/16	３５０　Ｆ	8840－5L		
9/46	３６０　Ｂ	8120－5B		
12/00	５７２	8841－5B		

審査請求　未請求　請求項の数３(全　７　頁)

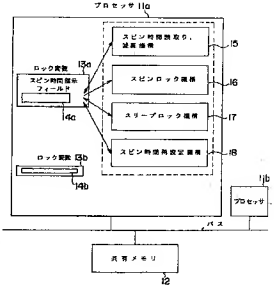
(21)出願番号　特願平4－25619

(22)出願日　平成４年(1992)２月13日

(71)出願人　000003078
株式会社東芝
神奈川県川崎市幸区堀川町72番地
(72)発明者　岐津　俊樹
神奈川県川崎市幸区小向東芝町１番地　株式会社東芝総合研究所内
(74)代理人　弁理士　鈴江　武彦

(54)【発明の名称】　ロック方式

(57)【要約】
【目的】ロック獲得の際のプロセッサの浪費を少なくし、プロセス実行の効率化の向上を図る。
【構成】ロック変数１３ａのロックを獲得できない場合には、そのプロセスはスピンロック機構１６によってスピンロック状態に設定され、そして所定のスピンロック時間スピンしてもロックを獲得できない場合にスリープロック機構１７によってスリープ状態に設定される。このため、スピン時間指示フィールドに設定された所定のスピンロック時間を越える長い時間ロックが取れない時にはスリープとなって、プロセッサを他のプロセスに渡すことができ、また、ロックが他のプロセスから比較的早く解放された場合にはスピンロック状態にあるので直ぐにそのロックを獲得して実行することができる。



1

【特許請求の範囲】

【請求項1】　共有変数を用いてプロセス間の排他制御を実現するメモリ共有型マルチプロセッサシステムにおいて、

前記各共有変数毎にロック待ちプロセスがスピンするスピンロック時間を規定し、

ロックを獲得できなかったロック待ちプロセスを前記規定されたスピンロック時間だけスピンさせ、そのスピンロック時間内にロックを獲得できないときにそのプロセスをスリープさせることを特徴とするマルチプロセッサシステムのロック方式。

【請求項2】　前記スピンロック時間は、プロセスのスリープおよびそのスリープ状態からの復帰のために要するプロセッサの実行時間と実質的に等しい値に初期設定されていることを特徴とする請求項1記載のマルチプロセッサシステムのロック方式。

【請求項3】　共有変数を用いてプロセス間の排他制御を実現するメモリ共有型マルチプロセッサシステムにおいて、

前記各共有変数毎にロック待ちプロセスがスピンするスピンロック時間を割り当てる手段と、

ロックを獲得できなかったプロセスを前記スピンロック時間だけスピンさせる手段と、

ロック解除待ちプロセスを有する共有変数に対応した前記スピンロック時間の値を減少する手段と、

前記スピンロック時間内にロックを獲得できないときにそのプロセスをスリープさせる手段と、

ロック獲得後そのロックを解放する際に前記スピン時間を再設定する手段とを具備したことを特徴とするマルチプロセッサシステムのロック方式。

【発明の詳細な説明】

【０００１】

【産業上の利用分野】この発明は高性能な並列処理を実現するための共有メモリを有したマルチプロセッサシステムの排他制御におけるロック方式に関するものである。

【０００２】

【従来の技術】近年、共有メモリを有したマルチプロセッサシステムにおいては、排他制御のための種々のロック方式が提案されている。ロック方式には、大別してプロセスがロックを取れるまで共有変数（ロック変数）にアクセスをし続けるスピンロック方式と、ロックが取れなかった場合スリープし、ロックが解放された時に起こされてからロックを取りに行くスリープロック（サスペンドロックと称する場合もある）方式とがある。

【０００３】スピンロック方式では、スピンしている期間はプロセッサは浪費されることになる。一方、スリープロック方式では、プロセスのスリープとそのスリープからの復帰（ウェイクアップ）の際にレジスタ退避やレジスタ復帰が必要になるため、プロセッサ（ＣＰＵ）実

2

行時間が余分にかかりその分だけコストが大きくなるが、プロセスがスリープしている間はプロセッサは他のプロセスの実行に使用でき、プロセッサが浪費されることはない。

【０００４】従来のロック方式では、資源がロックされている時間がわからないため、予めスピンするかスリープするか決められており、長い時間ロックが取れない時にスピンしたり、ロックがすぐ取れる時にスリープするなど、プロセッサを浪費してしまうという問題があった。

【０００５】

【発明が解決しようとする課題】従来のロック方式においては、ロックを取得する際に予めスピンするかスリープするかを決定しておくが、資源がロックされる時間がわからないために無駄にスピンしたりスリープしてしまうという問題があった。

【０００６】この発明はこのような点に鑑みてなされたものであり、ロック獲得の際のプロセッサの浪費を少なくし、効率の良いプロセスの実行の実現することができるマルチプロセッサシステムのロック方式を提供することを目的とする。

【０００７】

【課題を解決するための手段および作用】この発明のロック方式は、共有変数を用いてプロセス間の排他制御を実現するメモリ共有型マルチプロセッサシステムにおいて、前記各共有変数毎にロック待ちプロセスがスピンするスピンロック時間を規定し、ロックを獲得できなかったロック待ちプロセスを前記規定されたスピンロック時間だけスピンさせ、そのスピンロック時間内にロックを獲得できないときにそのプロセスをスリープさせることを特徴とする。

【０００８】このロック方式においては、ロックを獲得できない場合には、そのプロセスはまずスピンロック状態に設定され、そして、所定のスピンロック時間スピンしてもロックを獲得できない場合にスリープ状態に設定される。このため、所定のスピンロック時間を越える長い時間ロックが取れない時にはスリープとなって、プロセッサを他のプロセスに渡すことができ、また、ロックが他のプロセスから比較的早く解放された場合にはスピンロック状態にあるのですぐにそのロックを獲得して実行することができる。したがって、不必要に長い時間スピンしたり、短い時間スピンすればロックが獲得できるのにスリープするといったプロセッサの浪費が避けられ、プロセスの効率的な実行が可能となる。

【０００９】

【実施例】以下、図面を参照してこの発明の実施例を説明する。

【００１０】図1にはこの発明の一実施例に係わるマルチプロセッシステムのシステム構成が示されている。このマルチプロセッサシステムは、メモリ共有型のマル

3

チプロセッサであり、共有変数を用いて排他制御を実現している。

【００１１】このマルチプロセッサシステムは、複数のプロセッサ１１ａ，１１ｂ，…と、共有メモリ１２とから構成されている。複数のプロセッサ１１ａ，１１ｂ上で並列動作される複数のプロセスは、共有メモリ１２上の共有データを同時に操作しないように、その共有データをロックして排他的に使用する。この排他制御のための同期は、共有変数（ロック変数）を用いて実現されている。

【００１２】ロック変数はあるリソースの使用状態を示すものであり、例えば、ロック変数の値が１の時は、ある１つのプロセスが排他的な処理を実行中であること（ロック状態）を示し、他のプロセスはその処理を実行することができない。

【００１３】ロック確保時のロック変数の参照、更新は、例えば、テストアンドセット命令（ｔｅｓｔ　＆　ｓｅｔ）のように参照と変更が不可分に行われるプロセッサ命令で行なわれる。テストアンドセット命令は、基本的には不可分の単一機会命令として、指定されたロック変数から値"０"を読み出した後、"１"の値をロック変数に書き戻すものであり、最初に"０"（ロックがオフ）を受け取るプロセスだけがロックを確保して処理を実行でき、それ以外のプロセスはすべて"１"（ロックがオン）を受け取ることにより実行を待たされる。

【００１４】この場合、スピンロック方式においては、"１"を受け取った各プロセスは、ロック変数が"０"になるまで繰り返しテストを行い、ロック解除されまで待つ。この状態は、スピンループと称されている。一方、スリープロック方式においては、"１"を受け取った各プロセスは、プロセッサを解放するスリープ状態となり、ロック解除の合図を待ってウェイクアップされる。

【００１５】ロック変数１３ａ，１３ｂ…は、共有メモリ１２上の共有データの排他制御のためにそれら共有データ毎に生成されるものであって、各ロック変数１３ａ，１３ｂ，…にはスピン時間指示フィールド１４ａ，１４ｂ，…が割り当てられている。

【００１６】各スピン時間指示フィールド１４ａ，１４ｂには、対応するロック変数によってロック待ちとなったプロセスに対してスピンすべき時間を指示するためのスピン時間が設定される。このスピン時間の初期値は、スピンし続けるコストと、スリープし、ウェイクアップするコストとが等しくなるような値に設定される。

【００１７】この場合、スリープし、ウェイクアップするコストとは、プロセスのスリープおよびウェイクアップのために必要なレジスタ退避および復帰に要するプロセッサ実行時間（実行サイクル数）を示す。このため、スピン時間は、プロセスのスリープおよびウェイクアップのために必要なプロセッサ実行時間と実質的に同じ値

4

に予め設定される。これにより、スピンしている間にロックが獲得できた場合には、スリープするよりも低コストになることが保証される。

【００１８】また、各プロセッサ１１Ａ，１１ｂ，…には、ロック変数１３ａ，１３ｂ，…を用いたロック方式を実現するための機構として、スピン時間読取り・減算機構１５、スピンロック機構１６、スリープロック機構１７、スピン時間再設定機構１８が設けられている。

【００１９】次に、図２のフローチャートを参照して、これらスピン時間読取り・減算機構１５、スピンロック機構１６、スリープロック機構１７、およびスピン時間再設定機構１８の機能、およびこれらを利用したロック動作を説明する。

【００２０】プロセスが資源を確保する際には、まず、該当するロック変数１３ａおよびそのロック変数に対応するスピン時間フィールド１４ａが参照され、スピン時間読取り・減算機構１５によって上記フィールドの値を記憶した上で（ステップａ１）そのフィールドの値が減少される（ステップａ２）。

【００２１】減少される値は任意の値で良く、ここではｄ１とする。これは次にロックを取り に来るプロセスはロックを取りにくくなっており、スピンする時間を短くすることによってスピンによるプロセッサの浪費を減少させるためである。

【００２２】次に、ロック変数の値が"０"か"１"かによってロックを確保できるか否かを判断し（ステップａ３）、ロック変数の値が"１"の場合には、そのプロセスはロックを確保できない。このロックを確保できなかったプロセスはスピンロック機構１６によってスピンされ、ステップａ１で読み取ったスピン時間だけスピンしてその間、繰り 返しロックを獲得しに行く（ステップａ３，ａ４）。

【００２３】スピン時間内にロックが獲得できなかった場合には、さらに上記スピン時間指定フィールドの値がスリープロック機構１７によって減少され（ステップａ５）、そして、ロックが獲得できなかったプロセスがスリープロック機構１７によってスリープされる（ステップａ６）。この時に減少させる値も任意の値で良く、ここではｄ２とする。

【００２４】この場合に、上記フィールドの値を再び減らすのは、スピン時間中にロックが取れないほどロックが獲得しにくくなっているため、さらにスピンする時間を短くすることによってスピンによる他のプロセッサの浪費を減少させるためである。

【００２５】スリープされたプロセスはロックが解放されたときに通知等によってウェイクアップされ（ステップａ７）、ロックを確保しに行く。ここでロックを確保できた場合には、そのプロセスによって所定の作業が実行される（ステップａ９）。この後、スピン時間再設定機構１８によって、ロックを獲得するまでに減じた値分

だけ上記フィールドの値が増加される（ステップa１０）。

【００２６】この例では、スピンしている間にロックが獲得できた場合には、ｄ１を、スピンしている間にロックが獲得できずスリープした後に獲得した場合には、ｄ１＋ｄ２を、上記フィールドの値に加算して初期値に戻す。

【００２７】このように上記フィールドの値を増加させるのは、ロックが解放されたことによりロックが解放される前に比べ獲得しやすくなるため、スピンする時間を長くしてスピンによりロックを獲得しやすくするためである。その後、プロセスは、ロックを解放する（ステップa１１）。次いで、図3のフローチャートを参照して、図1のシステムにおける別のロック動作を説明する。

【００２８】プロセスが資源を確保する際には、まず、該当するロック変数１３ａおよびそのロック変数に対応するスピン時間フィールド１４ａが参照され、スピン時間読取り・減算機構１５によって上記フィールドの値を記憶した上で（ステップb１）そのフィールドの値が減少される（ステップb２）。

【００２９】減少される値は任意の値で良く、ここではｄ１とする。これは次にロックを取りに来るプロセスはロックを取りにくくなっており、スピンする時間を短くすることによってスピンによるプロセッサの浪費を減少させるためである。

【００３０】また、同時に時刻を記録するフィールドに現在の時刻を、スピン時間読取り・減算機構１５によって記録しておく（ステップb３）。次に、ロック変数の値が“０”か“１”かによってロックを確保できるか否かを判断し（ステップb４）、ロック変数の値が“１”の場合には、そのプロセスはロックを確保できず、スピンロック機構１６によってスピンされ、ステップb１で読み取ったスピン時間だけスピンしてその間、繰り返しロックを獲得しに行く（ステップb４，b５）。

【００３１】スピン時間内にロックが獲得できなかった場合には、さらに上記スピン時間指定フィールドの値がスリープロック機構１７によって減少され（ステップb６）、そして、ロックが獲得できなかったプロセスがスリープロック機構１７によってスリープされる（ステップb７）。この時に減少させる値も任意の値で良く、ここではｄ２とする。

【００３２】この場合に、上記フィールドの値を再び減らすのは、スピン時間中ではロックが取れないほどロックが獲得しにくくなっているため、さらにスピンする時間を短くすることによってスピンによる他のプロセッサの浪費を減少させるためである。

【００３３】スリープされたプロセスはロックが解放されたときに通知等によってウェイクアップされ（ステップb８）、ロックを確保しに行く。ここでロックを確保できた場合には（ステップb９）、スピン時間再設定機構１８によって、ロックをとれた時刻とスピンを開始した時刻との差により、ロック獲得に要した時間が求められる（ステップb１０）。また、そのロックを獲得したプロセスによって所定の作業が実行される（ステップb１１）。

【００３４】この後、スピン時間再設定機構１８によって、ロックを獲得するまでに減じた値（ｄ１，またはｄ１＋ｄ２）と、ロック獲得に要した時間と、ステップb１で記憶したもとのスピン時間とから、新たなスピン時間が求められ、それがスピン時間指示フィールドに再設定される（ステップb１２）。そして、その後、ロックが解放される（ステップb１３）。この再設定される新たなスピン時間x´は、例えば、次のように求められる。

【００３５】x′を新しいスピン時間、xを読み取ったスピン時間、x０をスピン時間の初期値、yをロックを獲得するまでにかかった時間、dをロックを獲得するまでに減算した値、nを任意の正数とすると、

$$x' = \{(n+1)/n\}x - (y/n) + d \qquad \cdots(1)$$

で与えられる。しかし、この（１）式で求めたx´は、求めたx´が$0 < x' < x_0$の範囲にある場合のみ有効とし、（１）式で求めたx´が$x' \geqq x_0$の場合には、

$$x' = x_0 \qquad \cdots(2)$$

とし、また、（１）式で求めたx´が$x' \leqq 0$の場合には、

$$x' = 0 \qquad \cdots(3)$$

とすることが好ましい。

【００３６】これにより、スピンする時間は上記のnの大小によって振れ幅が変化するが、徐々にその資源を獲得するまでの時間の平均値に収束していく。これにより資源を獲得する際には、実際にロック獲得までに要した平均的な時間分だけスピンするようになり、プロセッサの浪費が減少することになる。

【００３７】以上のように、この実施例のロック方式においては、ロックを獲得できない場合には、そのプロセスはまずスピンロック状態に設定され、そして、所定のスピンロック時間スピンしてもロックを獲得できない場合にスリープ状態に設定される。このため、所定のスピンロック時間を越える長い時間ロックが取れない時にはスリープとなって、プロセッサを他のプロセスに渡すことができ、また、ロックが他のプロセスから比較的早く解放された場合にはスピンロック状態にあるのですぐにそのロックを獲得して実行することができる。

【００３８】したがって、不必要に長い時間スピンした

7

り、短い時間スピンすればロックが獲得できるのにスリープするといったプロセッサの浪費が避けられ、プロセスの効率的な実行が可能となる。

【００３９】また、スピンする時間を静的、あるいは動的に変更することにより、ロックされている時間の変動に柔軟に対処でき、効率の良いプロセスの実行が可能となるなど多大なる効果が得られる。

【００４０】

【発明の効果】以上説明したようにこの発明によれば、ロック獲得の際のプロセッサの浪費を少なくでき、効率の良いプロセスの実行の実現することが可能となる。

【図面の簡単な説明】

8

【図１】この発明の一実施例に係わるシステム構成を示すブロック図。

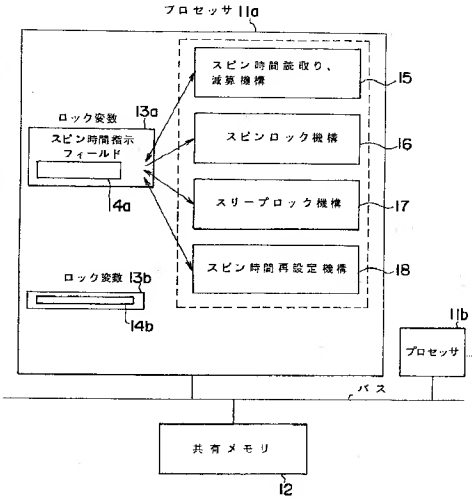
【図２】同実施例のロック動作の一例を説明するフローチャート。

【図３】同実施例のロック動作の他の例を説明するフローチャート。

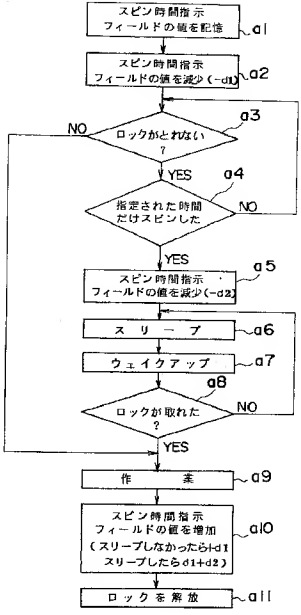
【符号の説明】

１１ａ，１１ｂ…プロセッサ、１２…共有メモリ、１３ａ，１３ｂ…ロック変数、１４ａ，１４ｂ…スピン時間指示フィールド、１５…スピン時間読取り・減算機構、１６…スピンロック機構、１７…スリープロック機構、１８…スピン時間再設定機構。

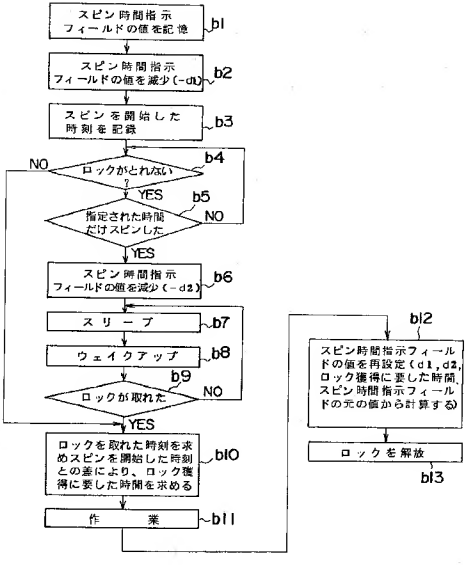
【図１】



【図2】


スピン時間指示
フィールドの値を記憶
a1
スピン時間指示
フィールドの値を減少(−d1)
a2
a3
NO
ロックがとれない
?
YES
a4
指定された時間
だけスピンした
NO
YES
a5
スピン時間指示
フィールドの値を減少(−d2)
スリープ
a6
ウェイクアップ
a7
a8
ロックが取れた
?
NO
YES
作業
a9
スピン時間指示
フィールドの値を増加
(スリープしなかったらd1
スリープしたらd1+d2)
a10
ロックを解放
a11

【 図3 】


スピン時間指示フィールドの値を記憶
b1
スピン時間指示フィールドの値を減少(－d1)
b2
スピンを開始した時刻を記録
b3
ロックがとれない？
b4
NO
YES
指定された時間だけスピンした
b5
NO
YES
スピン時間指示フィールドの値を減少(－d2)
b6
スリープ
b7
ウェイクアップ
b8
ロックが取れた
b9
NO
YES
ロックを取れた時刻を求めスピンを開始した時刻との差により、ロック獲得に要した時間を求める
b10
作業
b11
b12
スピン時間指示フィールドの値を再設定(d1,d2,ロック獲得に要した時間,スピン時間指示フィールドの元の値から計算する)
ロックを解放
b13